OIPE JC24
NOV 02 2004
PATENT & TRADEMARK OFFICE

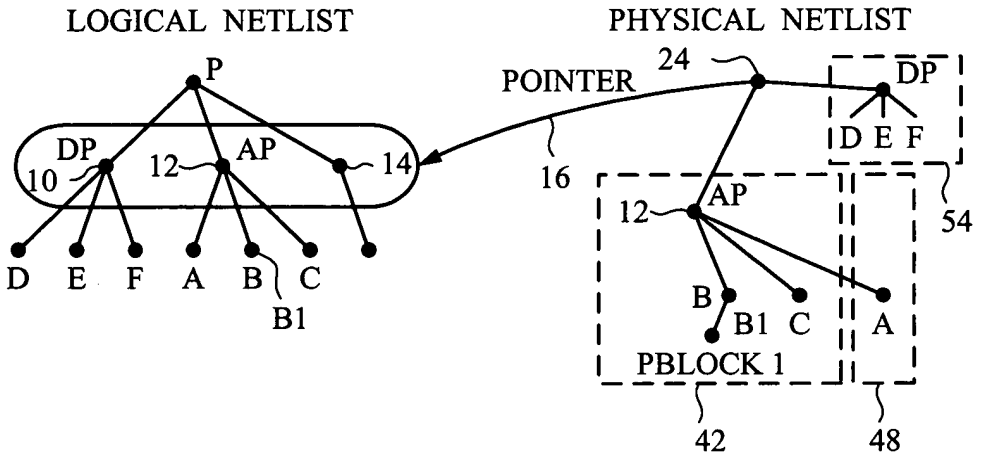

FIG. 1

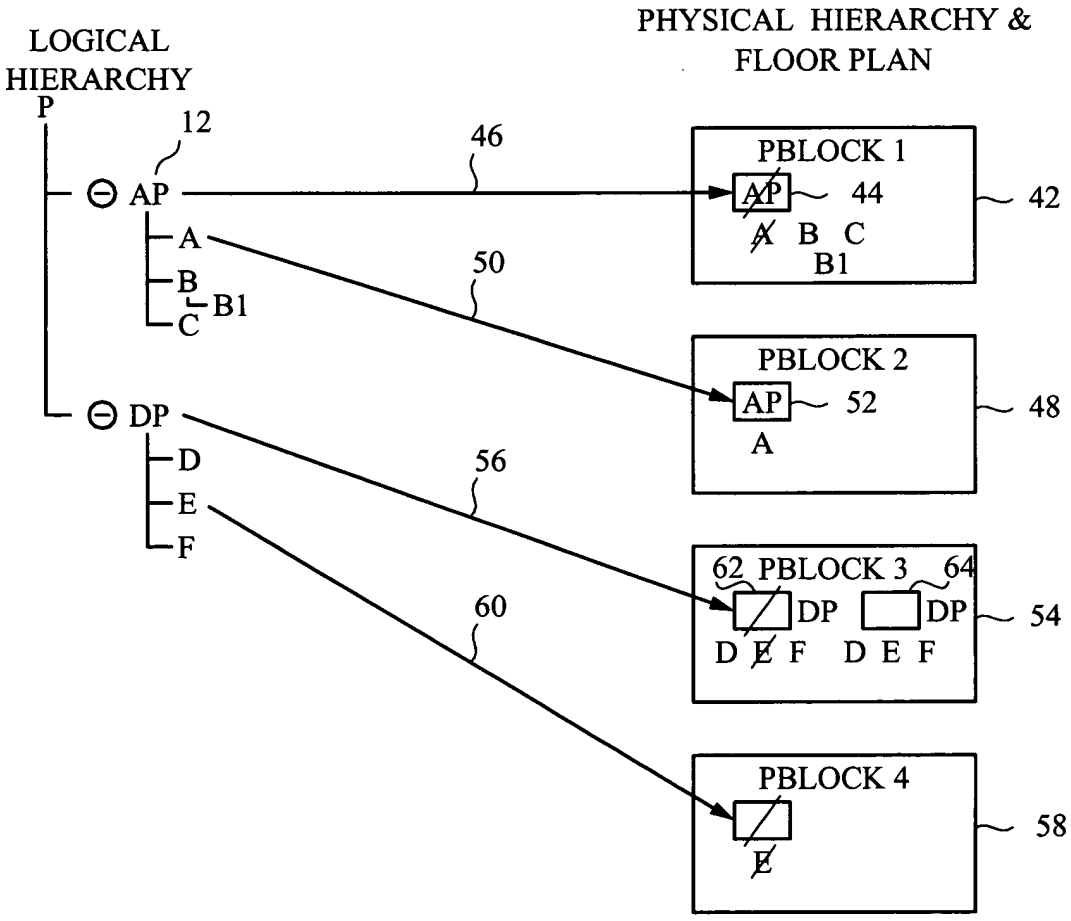

FIG. 5

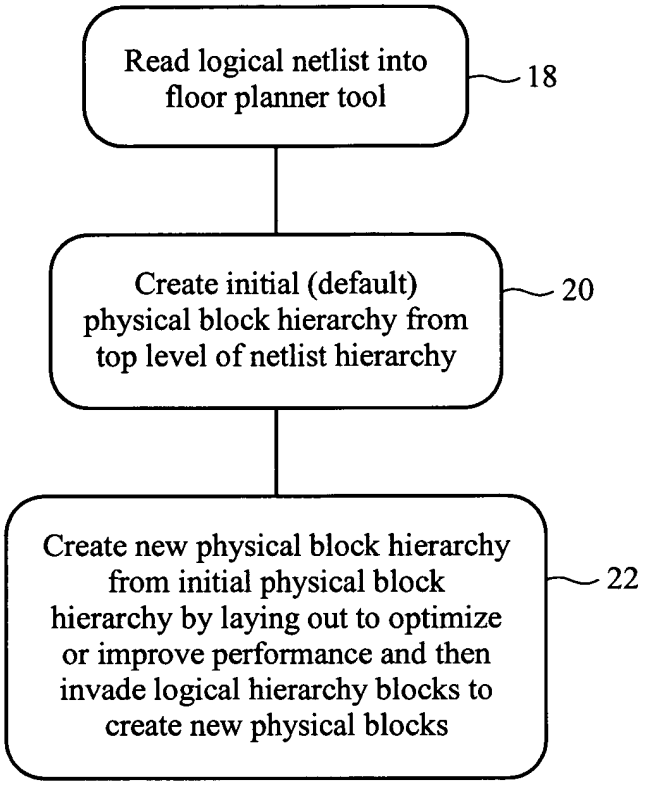
Read logical netlist into floor planner tool
18
Create initial (default) physical block hierarchy from top level of netlist hierarchy
20
Create new physical block hierarchy from initial physical block hierarchy by laying out to optimize or improve performance and then invade logical hierarchy blocks to create new physical blocks
22

FIG. 2

PROCESS TO CREATE A NEW FLOORPLAN USING IMPROVED FLOOR PLANNER

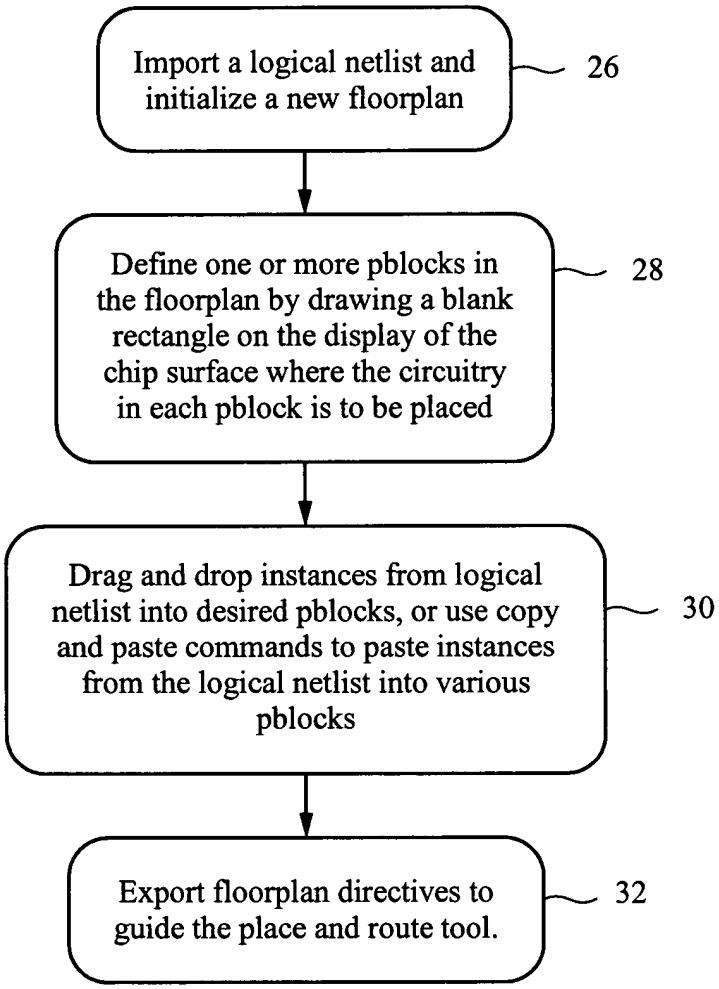

FIG. 3

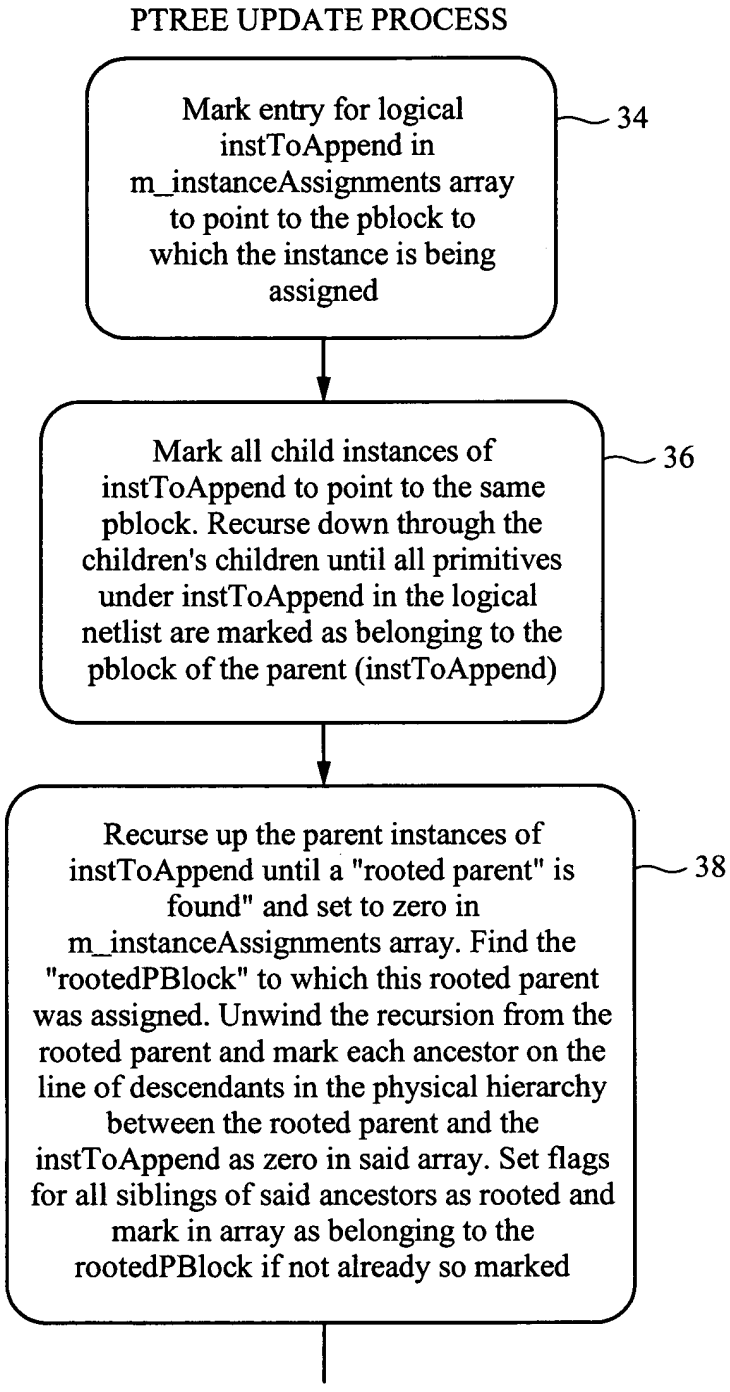
PTREE UPDATE PROCESS
Mark entry for logical instToAppend in m_instanceAssignments array to point to the pblock to which the instance is being assigned
34
Mark all child instances of instToAppend to point to the same pblock. Recurse down through the children's children until all primitives under instToAppend in the logical netlist are marked as belonging to the pblock of the parent (instToAppend)
36
Recurse up the parent instances of instToAppend until a "rooted parent" is found" and set to zero in m_instanceAssignments array. Find the "rootedPBlock" to which this rooted parent was assigned. Unwind the recursion from the rooted parent and mark each ancestor on the line of descendants in the physical hierarchy between the rooted parent and the instToAppend as zero in said array. Set flags for all siblings of said ancestors as rooted and mark in array as belonging to the rootedPBlock if not already so marked
38

FIG. 4A

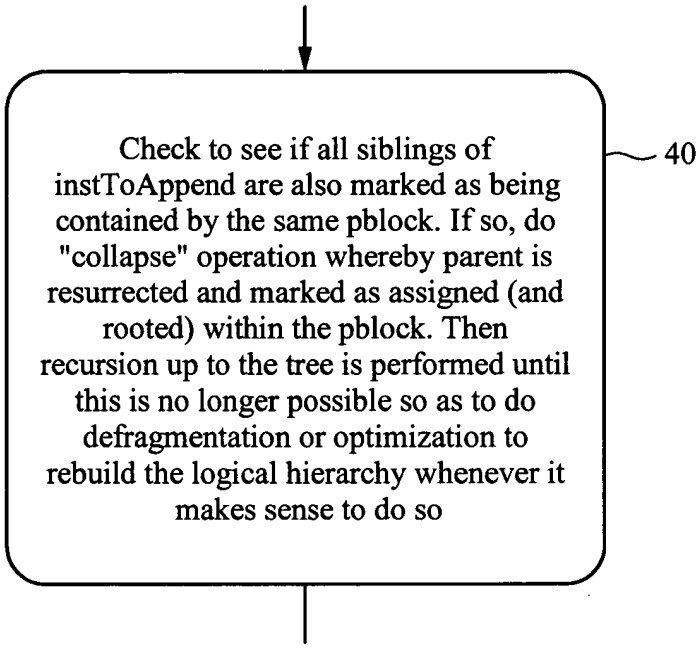
Check to see if all siblings of instToAppend are also marked as being contained by the same pblock. If so, do "collapse" operation whereby parent is resurrected and marked as assigned (and rooted) within the pblock. Then recursion up to the tree is performed until this is no longer possible so as to do defragmentation or optimization to rebuild the logical hierarchy whenever it makes sense to do so
40

FIG. 4B

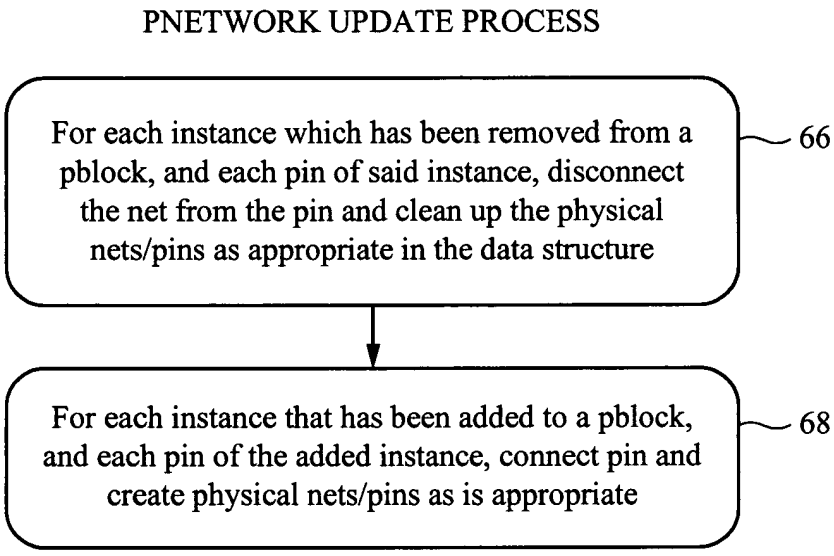
PNETWORK UPDATE PROCESS
For each instance which has been removed from a pblock, and each pin of said instance, disconnect the net from the pin and clean up the physical nets/pins as appropriate in the data structure
66
For each instance that has been added to a pblock, and each pin of the added instance, connect pin and create physical nets/pins as is appropriate
68

FIG. 6

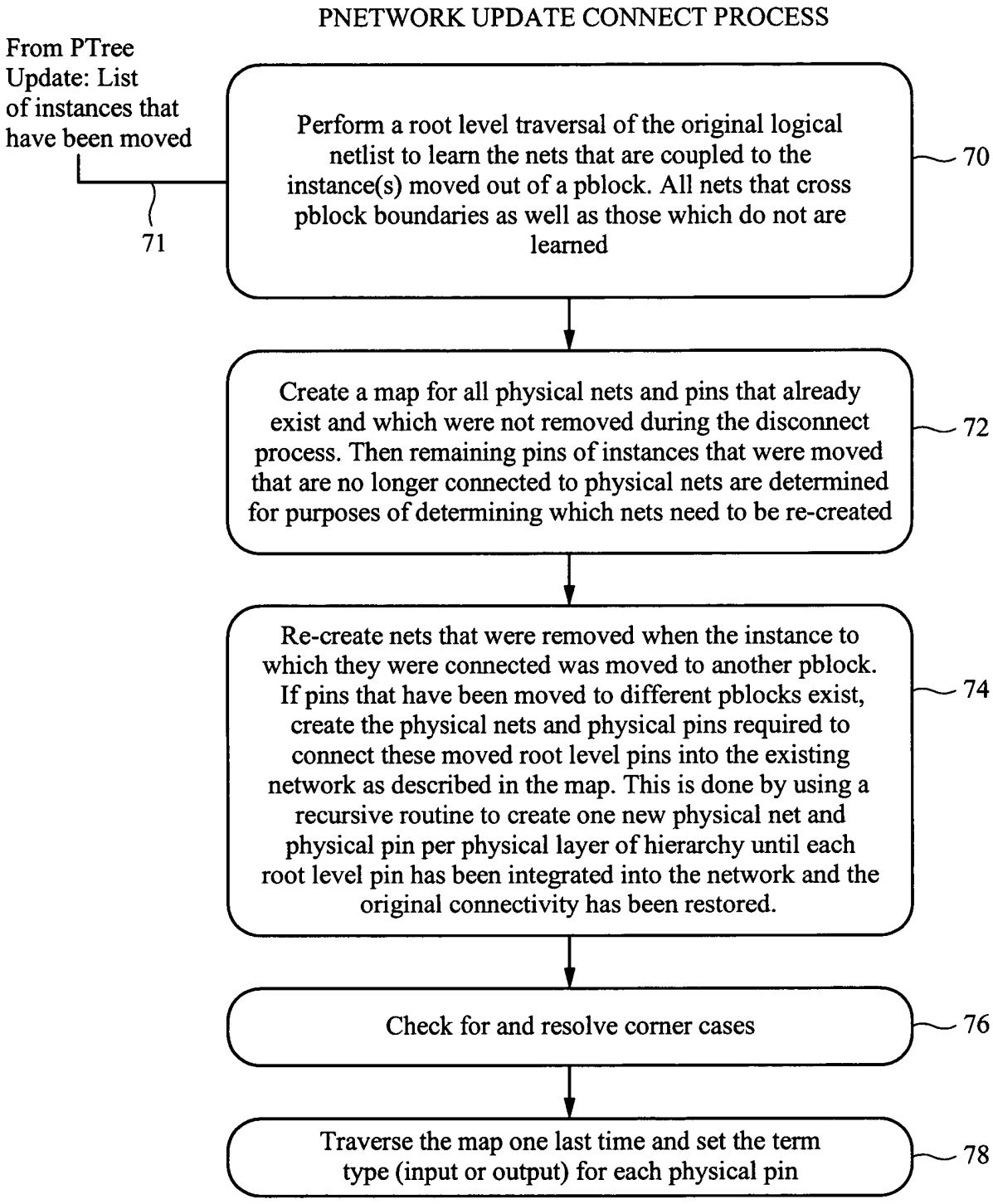
PNETWORK UPDATE CONNECT PROCESS
From PTree Update: List of instances that have been moved
71
Perform a root level traversal of the original logical netlist to learn the nets that are coupled to the instance(s) moved out of a pblock. All nets that cross pblock boundaries as well as those which do not are learned
70
Create a map for all physical nets and pins that already exist and which were not removed during the disconnect process. Then remaining pins of instances that were moved that are no longer connected to physical nets are determined for purposes of determining which nets need to be re-created
72
Re-create nets that were removed when the instance to which they were connected was moved to another pblock. If pins that have been moved to different pblocks exist, create the physical nets and physical pins required to connect these moved root level pins into the existing network as described in the map. This is done by using a recursive routine to create one new physical net and physical pin per physical layer of hierarchy until each root level pin has been integrated into the network and the original connectivity has been restored.
74
Check for and resolve corner cases
76
Traverse the map one last time and set the term type (input or output) for each physical pin
78

FIG. 7

THE DISCONNECT PROCESS OF PNETWORK UPDATE

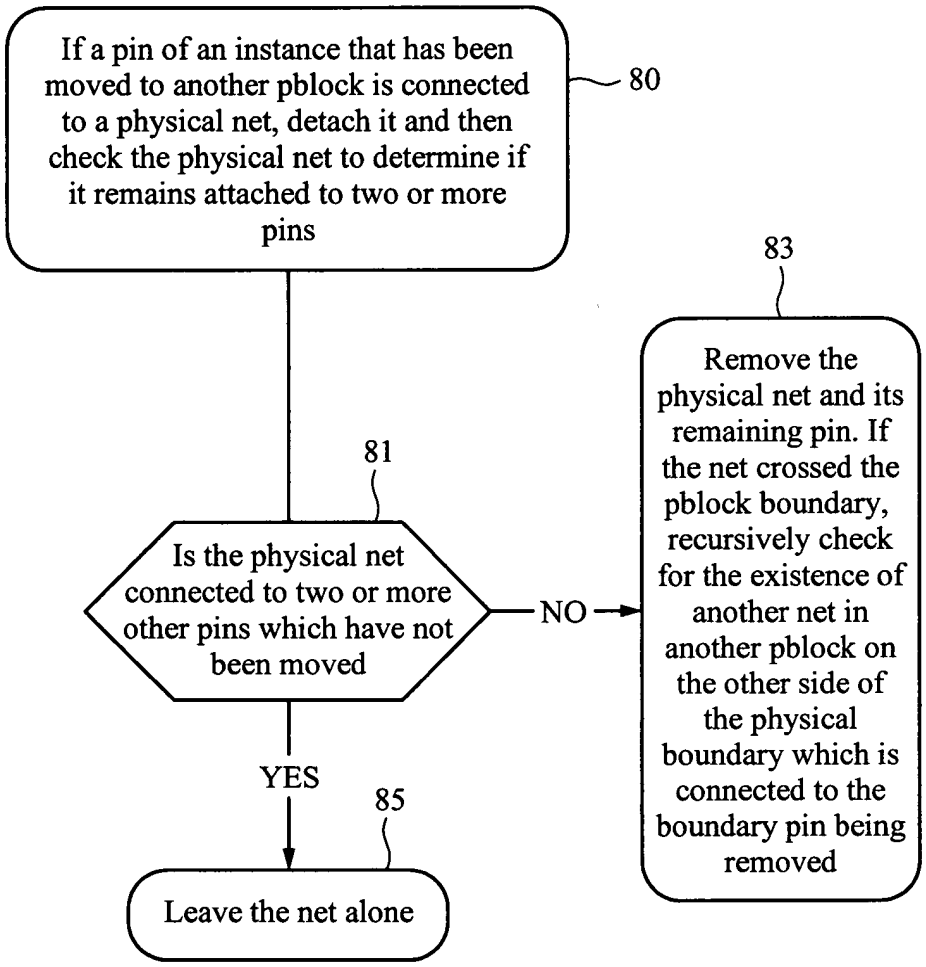

FIG. 8

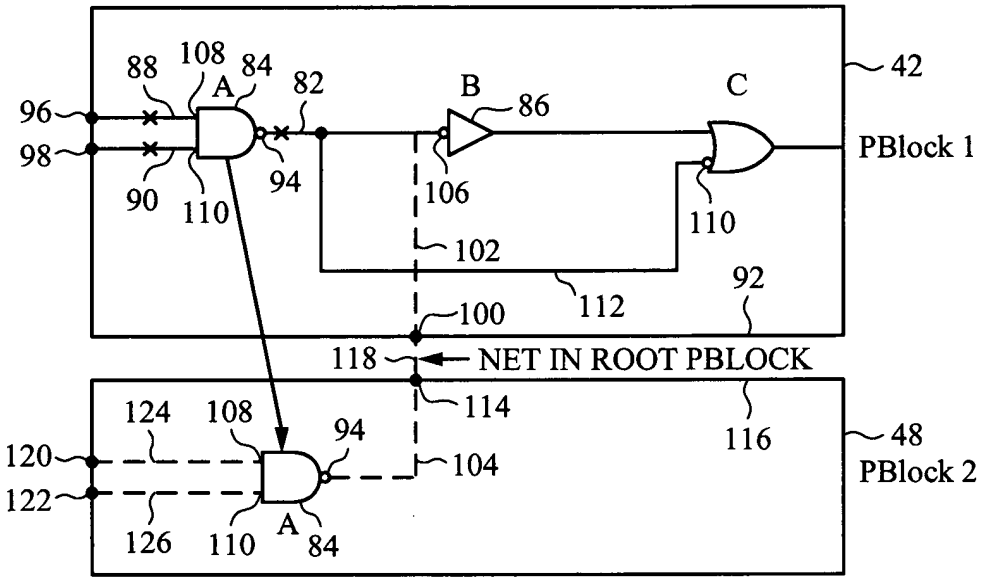
42
PBlock 1
88
108
84
A
82
B
86
C
96
98
90
110
94
106
110
102
112
92
100
118
NET IN ROOT PBLOCK
114
116
48
PBlock 2
124
108
94
120
122
104
126
110
A
84

FIG. 9

ALTERNATIVE EMBODIEMENT OF THE DISCONNECT PROCESS OF PNETWORK UPDATE

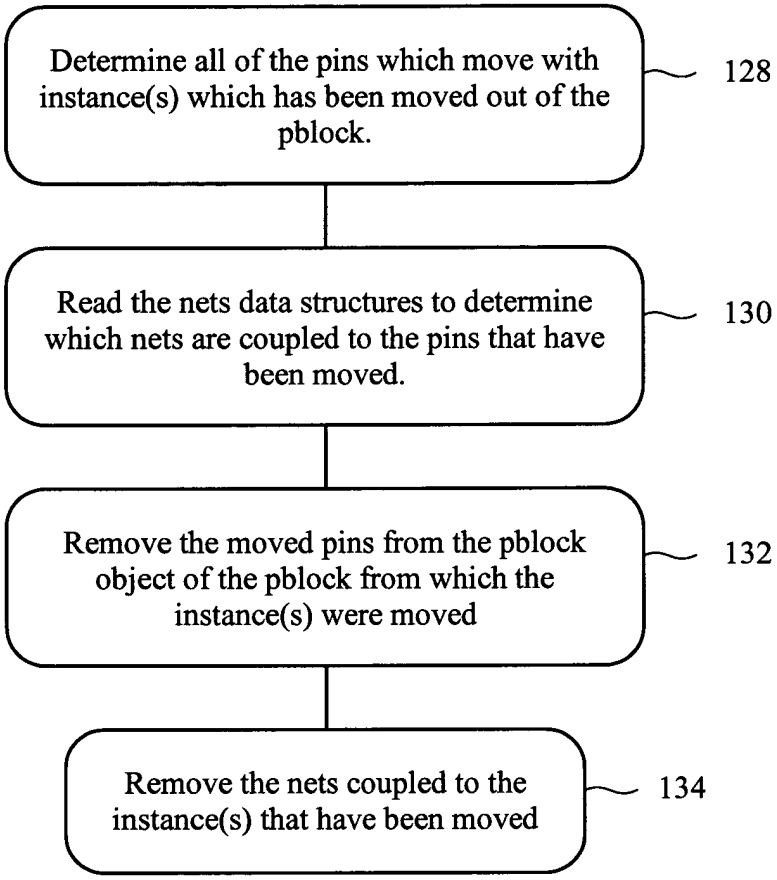

FIG. 10

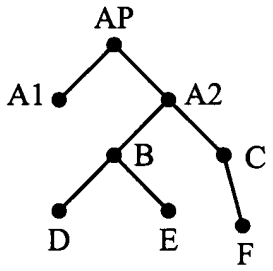

FIG. 11

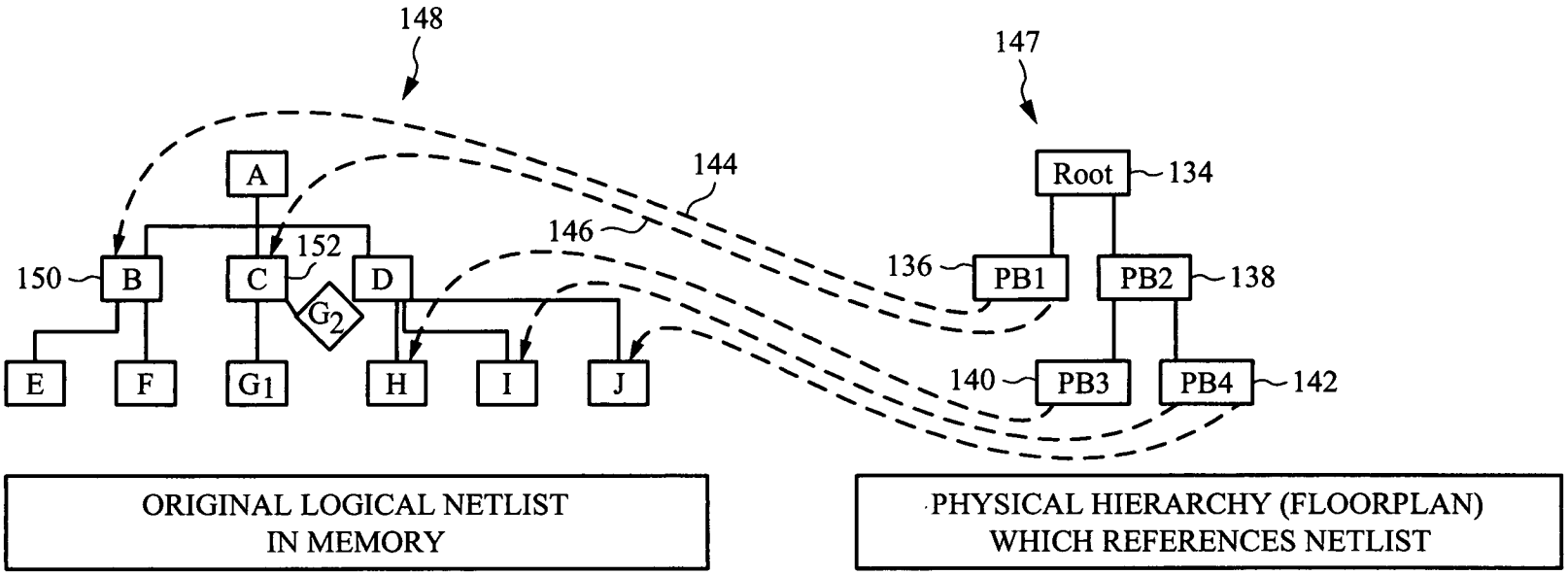
148
147
144
146
A
Root
134
150
B
152
C
D
136
PB1
PB2
138
G2
E
F
G1
H
I
J
140
PB3
PB4
142
ORIGINAL LOGICAL NETLIST
IN MEMORY
PHYSICAL HIERARCHY (FLOORPLAN)
WHICH REFERENCES NETLIST

FIG. 12

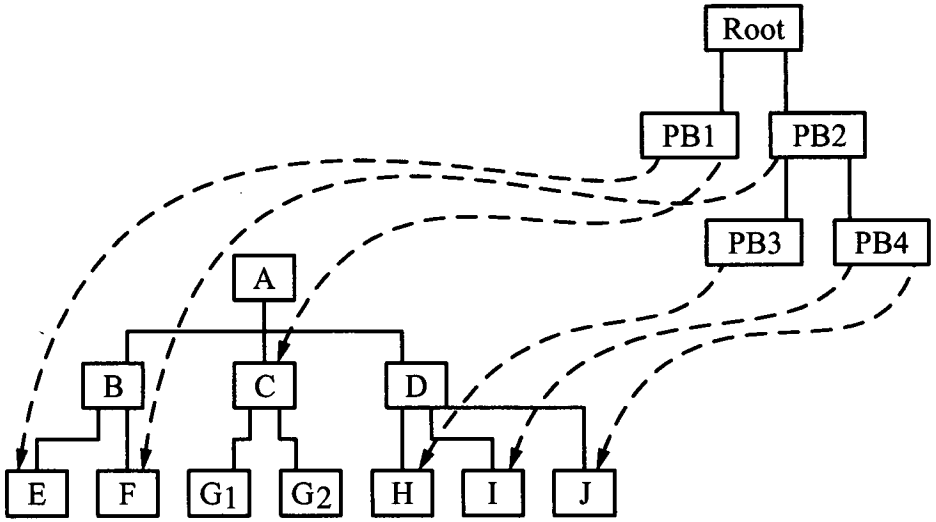
Root
PB1
PB2
PB3
PB4
A
B
C
D
E
F
G1
G2
H
I
J

FIG. 13

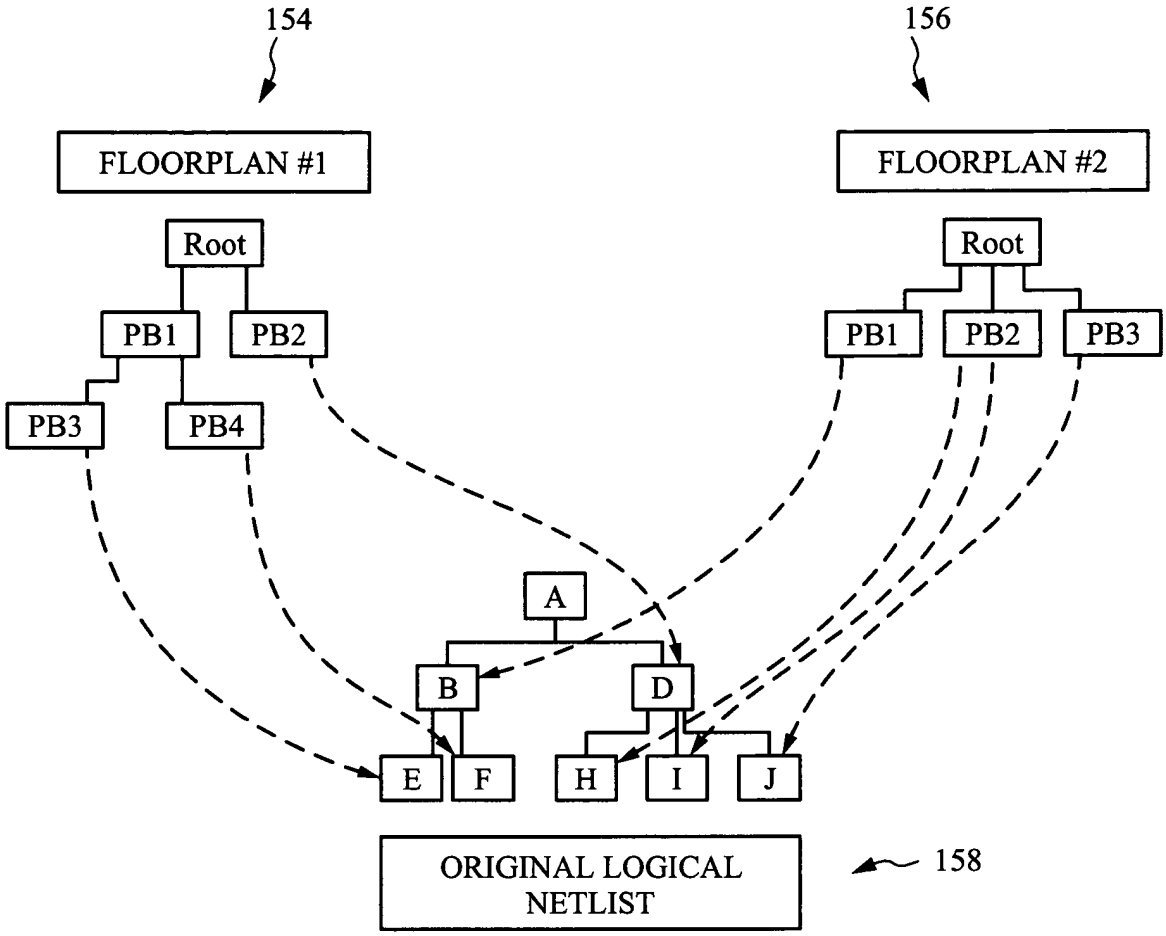
154
156
FLOORPLAN #1
FLOORPLAN #2
Root
PB1
PB2
PB3
PB4
Root
PB1
PB2
PB3
A
B
D
E
F
H
I
J
ORIGINAL LOGICAL NETLIST
158

FIG. 14

PROCESS TO DISCONNECT AND RECONNECT AN ENTIRE PBLOCK WHEN IT IS MOVED OUT OF ONE PARENT PBLOCK AND INTO ANOTHER

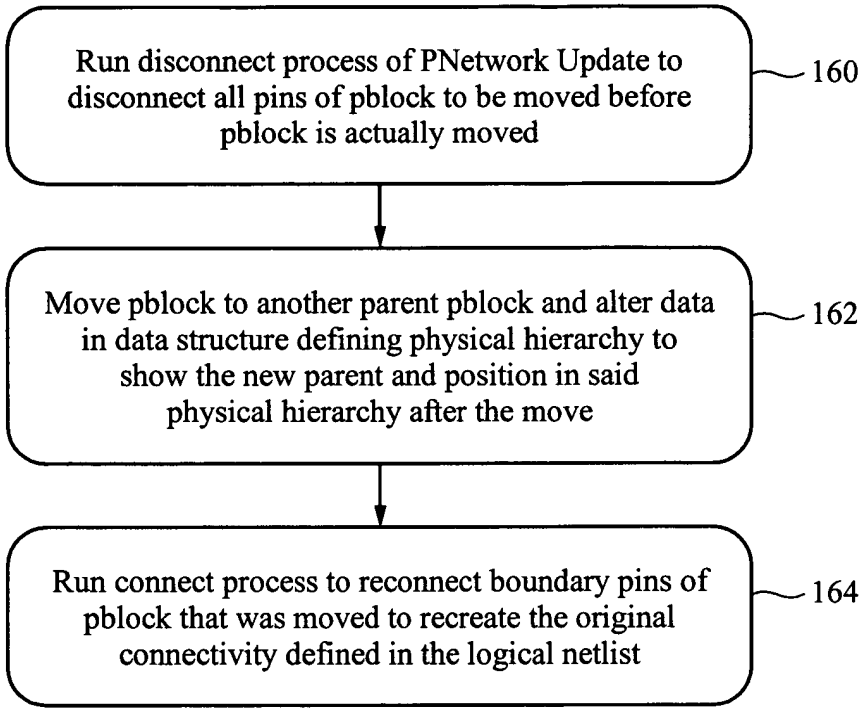

FIG. 15

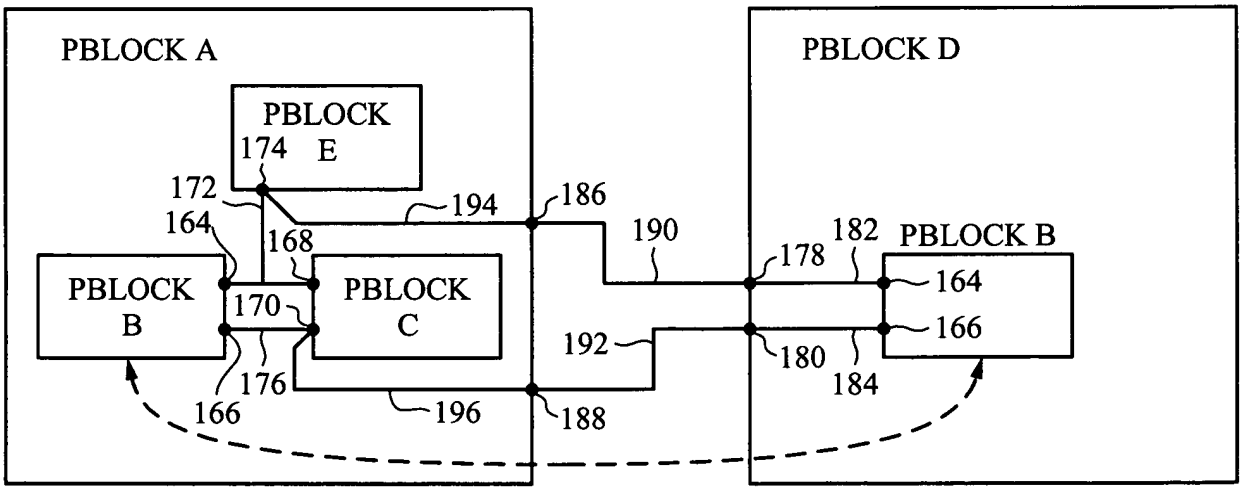

FIG. 16